NAWCWPNS TP 8248

Sparrow Test Set Improvements

by
Elden R. Sandy
Roger Geer
J. M. Zapata
Weapons and Targets Support Equipment Division
Support Equipment Department
and
Ira Lieberman
Test Automation, Inc.

MAY 1995

NAVAL AIR WARFARE CENTER WEAPONS DIVISION
POINT MUGU, CA 92042-5001

Approved for public release; distribution is unlimited.

Naval Air Warfare Center Weapons Division

FOREWORD

This report describes improvements to the Sparrow depot-level test set. This effort was conducted at the Naval Air Warfare Center Weapons Division, Point Mugu, Calif., by an engineering group composed of two branches in the Weapons and Targets Support Equipment Division, Code 486000E, and was sponsored by PMA-259 with weapon funding.

This report was reviewed for technical accuracy by W. J. Bertelt.

Under authority of
D. B. McKINNEY
RADM, U. S. Navy
Commander

Approved by
J. SANDOVAL, *Head*
Support Equipment Department
19 April 1995

Released for publication by
G. W. WROUT
Head, NAWCWPNS Test and Evaluation Group

NAWCWPNS Technical Publication 8248

Published by..Scientific and Technical Documentation
Collation ....Cover, 15 leaves
First printing..200 copies

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1995	Technical Publication

4. TITLE AND SUBTITLE

Sparrow Test Set Improvements (U)

5. FUNDING NUMBERS

6. AUTHOR(S)

Elden R. Sandy, Roger Geer, J. M. Zapata, and Ira Lieberman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Air Warfare Center Weapons Division
Point Mugu, CA 993402-5001

8. PERFORMING ORGANIZATION REPORT NUMBER

9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

PMA-259
Washington, D.C.

10 SPONSORING/MONITORING AGENCY REPORT NUMBER

NAWCWPNS TP 8248

11. SUPPLEMENTARY NOTES

12A. DISTRIBUTION/AVAILABILITY STATEMENT

A statement: approved for public release

12B. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

(U) This report describes improvements to the Sparrow depot-level test set, an inexpensive way to update the test set's technical manuals, and an innovative way to apprise sponsors of developments. Criteria for selecting test set components to be replaced were supportability, reliability, and maintainability.

14. SUBJECT TERMS

Sparrow	COTS	Basic language
Test set/station	ATE	Engineering Change Proposal
Technical Manuals	CASS	Work Breakdown Structure
	IADS	

15. NUMBER OF PAGES

29

16. PRICE CODE

10 SECURITY CLASSIFICATION OF REPORT	10 SECURITY CLASSIFICATION OF THIS PAGE	10 SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

Figures:

INTRODUCTION

U.S. naval forces have used the Sparrow AIM/RIM-7M/P air-to-air and surface-to-air missiles since the 1970s. The AN/DPM-22 Guided Missile Components Test Station dates from that era. At the core of the test station is the Hewlett Packard (HP)9500 Automated Test System, which is controlled by the HP2100S minicomputer and associated peripherals.

Although the test set had been reconfigured several times over the years, the HP2100S control computer (32K RAM, 5 MB disk size, and 1 MHZ clock speed) had finally reached its maximum capacity. Additionally, support and maintenance of the test set were an increasing problem, as was a high failure rate stemming from the obsolescent technology. An engineering team at Naval Air Warfare Center Weapons Division (NAWCWPNS) Point Mugu, Calif., was tasked with investigating how COTS equipment could improve reliability and supportability of the test set, as well as provide the extra control-computer memory needed to accommodate additional test capabilities for AIM-7P and Vertical Launch Sparrow. Criteria for selecting test set components to be replaced were supportability, reliability, and maintainability.

The aging Sparrow depot-level test set was updated at NAWCWPNS Point Mugu. At the same time an efficient and inexpensive way was devised to update the test set's technical manuals. The modernization effort emphasized the use of commercial off-the-shelf (COTS) equipment and used an innovative program structure and reporting system to keep the sponsor abreast of developments.

BACKGROUND

The Department of Defense (DOD) has a large investment in automatic test equipment (ATE), which has become increasingly difficult and costly to repair due to nonavailability of parts and trained maintenance personnel. One example is the AN/DPM-22 Guided Missile Components Test Station. A 1970-vintage test set based on the HP9500 Automated Test System.

To increase the service life of the AN/DPM-22, the Navy opted to replace the HP 2100S computer and its peripherals with an IBM-compatible 80486. One requirement was to retain the existing test equipment, which has an assortment of IEEE Standard 488 compatible and peculiar parallel interfaces. Another requirement was to minimize the software changes.

The AN/DPM-22 Guided Missile Components Test Station (Figure 1) was ATE developed for the Navy by the Raytheon Company in 1973. The test station is capable of performing depot-level tests of the guidance and control sections of the Sparrow missile,

versions AIM-7F and AIM/RIM-7M. The AN/DPM-22 was designed using the HP9500 Automated Test System as its core equipment. The HP9500 version in the AN/DPM-22 is controlled by the HP2100S minicomputer, associated disk drive, and other peripherals. Twenty-one years ago, the HP9500 with its HP2100S computer and peripherals was state-of-the-art. This equipment is costly to maintain, and may be unrepairable if any of the nonavailable components fail. Another Navy missile test set developed using the HP9500 is the AN/DSM-127 (Harpoon missile test set).

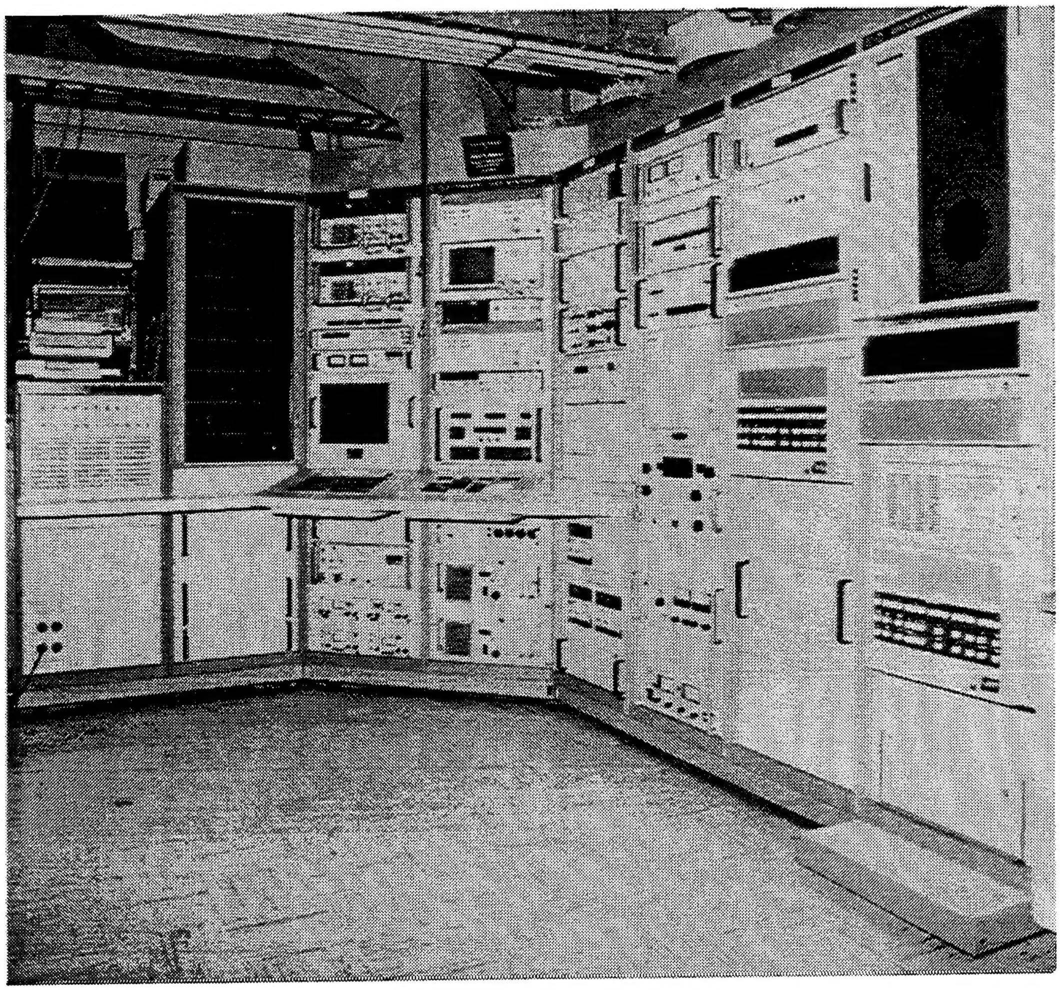

FIGURE 1. Sparrow AN/DPM-22 Guided Missile Components Test Station.

In 1992, a new version of the Sparrow missile, the AIM/RIM-7P, was developed and required testing at the repair depot. Because the Consolidated Automated Support System (CASS) missile test set was not yet available, the Naval Air Systems Command (NAVAIR) directed that the AN/DPM-22 be modified to incorporate AIM/RIM-7P tests.

Replacing no longer maintainable test set assemblies was a key to the success of this modification program. Analysis of repair records for the 13 AN/DPM-22 test sets suggested that replacing the HP2100S computer and its peripherals would provide the best return on investment. (Previous modifications had replaced some of the original HP9500 instruments with IEEE Standard 488, General Purpose Interface Bus (GPIB) compatible instruments.)

The task of redesigning and modifying four AN/DPM-22 test sets was assigned to the Weapons and Targets Support Equipment Division at Point Mugu. The process of replacing the computer and its peripherals follows.

DESIGN GOALS

We were tasked to replace obsolete equipment while striving to achieve the following design goals:

1. Maximize compatibility with the existing AN/DPM-22 design.

2. Use readily-available, multiple-source, off-the-shelf commercial components.

3. Select programming language compatible with the 500-plus existing test programs written in HP TODS BASIC.

4. Rewrite HP2100 assembly language subprograms in a high-level language for ease of software maintenance.

5. Create a paperless technical manual.

All aspects of the modification were to be documented. A set of engineering and logistics documentation for the new design was required.

HARDWARE DESIGN

The following original computer system equipment was removed from the AN/DPM-22 during the redesign and replaced by a new computer and peripherals.

1. HP2100S computer. Capabilities include 1 MHz clock; 32768 16-bit words of magnetic core memory; interrupt driven; direct memory access (DMA) capable. (DMA is not implemented in the AN/DPM-22).

2. HP7900A disk drive. Capabilities include 2.5M 16-bit words per disk; one fixed disk; one removable disk.

3. HP13215A disk drive power supply.

4. HP5321B-K21 digital clock. Capabilities include current date and time outputs; repetitive pulse generator with programmable pulse rate.

5. HP2645 display terminal. Capabilities include keyboard; 24 row by 80 column text display.

6. Beehive video display. Capabilities include 24 row by 80 column text display. Used to display messages to the operator when the HP2645 is not connected.

7. AN/DPM-22 system control panel. Capabilities include switches that provide operator input to the computer when the HP2645 is not connected.

8. AN/DPM-22 system display panel. Capabilities include display of measured data and test status using light emitting diode (LED) character displays and illuminated discrete indicators.

9. HP85001A (Dicom) tape cassette drive. Capabilities include storage of test data results.

The HP2155A input-output (IO) extender in the original computer system equipment was modified and retained in the AN/DPM-22. The IO extender is capable of increasing the maximum number of computer IO cards by 32 (see IO Extender Design).

The original computer system configuration is shown in Figure 2, the new configuration in Figure 3.

NEW SYSTEM DESIGN

A new IBM-PC-compatible, rack-mountable computer system was designed to perform the functions of the original computer system. The new computer includes the following components. Except as noted, all components selected are COTS.

1. Rack-mountable case with power supply for IBM-PC-compatible components.

2. Motherboard printed circuit assembly (PCA) with eight 16-bit ISA IO slots (three IO slots are also VESA Local Bus compatible), 80486 microprocessor with 33-MHz clock, 16 megabytes of random-access memory (RAM).

3. VESA Local Bus Windows accelerator and super-VGA video interface IO PCA.

4. Multifunction IO PCA. This VESA local bus compatible PCA provides two integrated drive electronics (IDE) hard drive interfaces, two floppy drive interfaces, two serial IO ports, and a parallel printer port.

5. GPIB interface IO PCA.

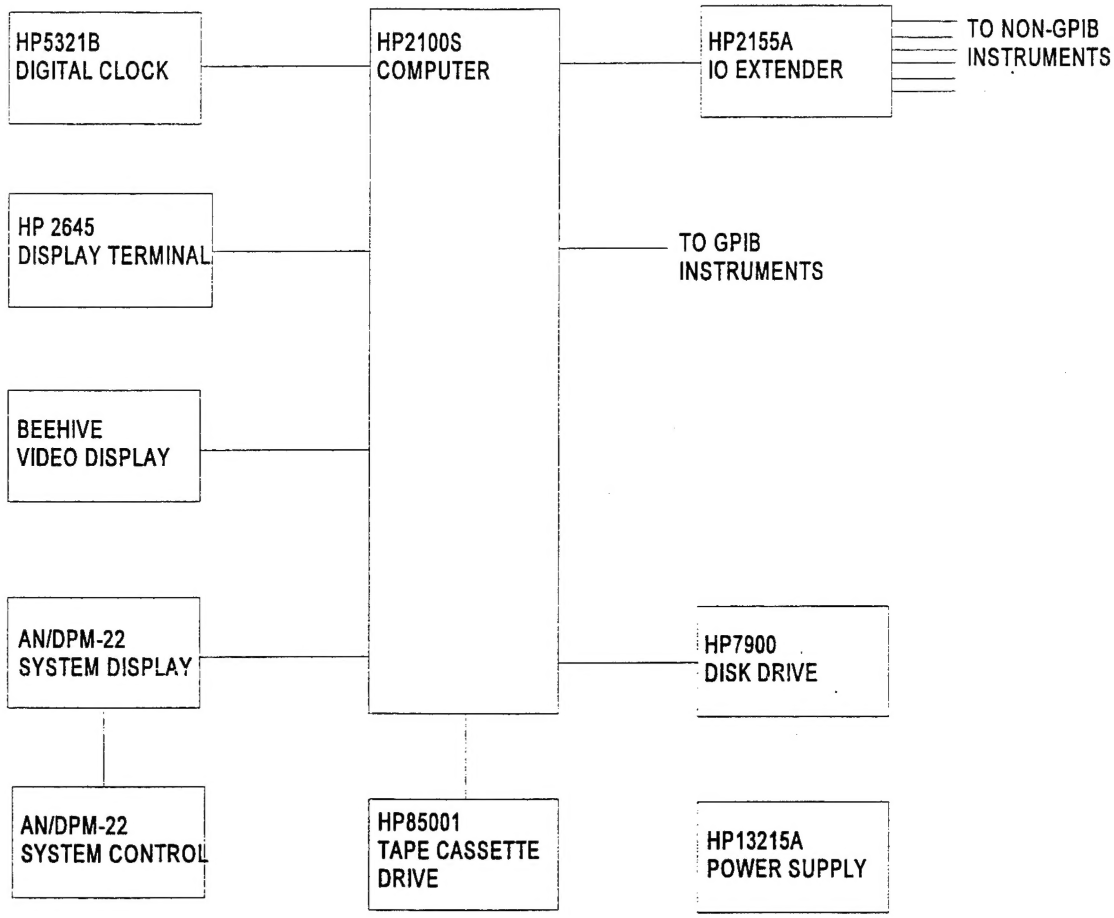

FIGURE 2. Original Configuration.

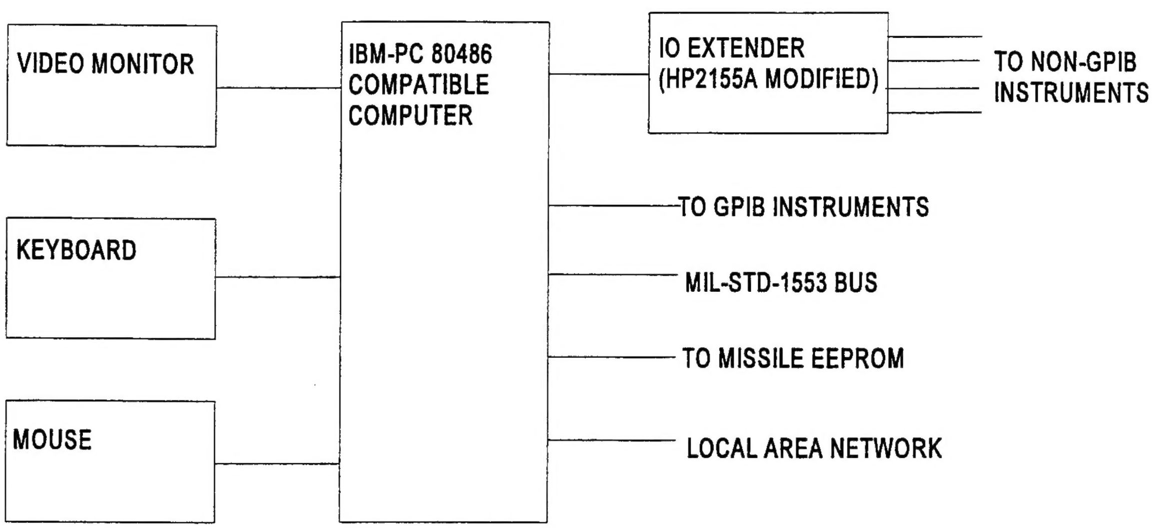

FIGURE 3. New Configuration.

6. Non-GPIB instrument interface IO PCA. This new PCA, the IOX486 PCA, was designed to provide interface between the new computer and the modified IO extender. The PCA provides a 16-bit-wide bidirectional data path, a 16-bit-wide IO-slot address path, four output data strobes, data valid status, and interrupt lines. The PCA also implements three timers that are used to provide accurate timing of test events to the nearest millisecond. Two groups of eight dual-in-line (DIP) switches are provided for user defined input to the computer. One switch bank is used for storing a unique test set hardware identification number. The IOX486 PCA appears to the 80486 as sixteen 16-bit IO ports.

Software was developed to control the IOX486 PCA (see Software Design). Cables connect the three connectors on this PCA to the three computer interface PCAs in the modified IO extender.

7. MIL-STD-1553 bus interface IO PCA (AIM/RIM-7P test requirement).

8. AIM/RIM-7P memory loader verifier interface IO PCA.

9. Local area network (LAN) interface IO PCA.

10. IDE hard disk drive with 540 megabytes.

11. Removable tray assembly for hard disk drive mounting.

12. Tape backup unit with 250 megabytes.

13. Keyboard.

14. Mouse with serial interface.

15. Rack-mountable case for desktop-style video monitor.

16. Super-VGA 17-inch color video monitor in desktop case.

DESIGN CRITERIA

The general selection criteria were availability of multiple sources of computer components, cost of components, and availability of operator and maintenance documentation. The following criteria were used when selecting specific computer components.

1. The 80486 motherboard PCA was selected to provide proven current technology. A clock speed of 33 MHz was chosen to achieve reliable microprocessor (MPU) operation without an MPU cooling fan.

An MPU cooling fan is a requirement when operating at clock rates of 66 MHz and above. When a fan is installed on the MPU, installing IO PCAs exceeding half-lengths into

two of the eight IO slots is not possible because the MPU socket is in line with the IO slots. This design trade-off was not compatible with the IO PCA configurations in the AN/DPM-22 computer.

2. RAM size of 16 megabytes was chosen to provide space for Microsoft Windows and the AN/DPM-22 paperless technical manual software, as well as growth potential for test programs.

3. The backup tape drive, a 250-megabyte model, was installed to provide a method of loading new software releases when the test stations are installed at the repair depot.

4. The LAN PCA was selected to be compatible with the facility's LAN. This compatibility allowed master software to reside on the LAN server and be downloaded into the test stations where the TPS software was being debugged.

INSTRUMENT INTERFACES

Instrument interfaces consist of the following types.

1. GPIB-compatible instruments. Seven GPIB-compatible instruments were originally connected to two HP GPIB interface IO PCAs installed in the HP2155A IO extender. The eight GPIB instruments in the new configuration are within the maximum GPIB interconnection distance allowed, so only one GPIB interface IO PCA is required. The instruments are connected to a commercial GPIB interface PCA installed in an IO slot of the new computer.

GPIB interfaces from two different manufacturers were investigated. However, one that had been used successfully in previous applications was not completely compatible with the VBDOS-PRO programming language (see Software Design). The National Instruments GPIB interface performed satisfactorily and was selected.

2. Non-GPIB Instruments. Each of the non-GPIB instruments was connected via cable to an HP IO PCA in the HP2155A IO extender (see IO Extender Design). These instruments have parallel digital interfaces. However, the interface designs are nonstandard, differing in the number of data, control, and status lines; logic voltage and current requirements; control, and status line signal definitions; and timing requirements. The type of IO PCA in the HP2155A IO extender was selected to accommodate the peculiar IO requirements of each instrument.

Eleven non-GPIB instruments are connected to new instrument interface PCAs in the modified IO extender (see IO Extender Design). No modifications to the instruments or their cables are required. Digital signals from the IO extender to the instruments have not changed.

IO Extender Design

The HP2155A original configuration has no processing capability. When connected to an HP2100S computer, the HP2155A IO extender provided the capability of adding up to 32 more instrument IO channels to the computer. The HP2155A contains 32 IO slots compatible with HP2100S IO PCAs, and three slots for PCAs that interface to the HP2100S.

The HP2100S control of and communication with the HP2155A is performed via two (or three) PCAs in the HP2100S and two (or three) PCAs in the HP2155A connected by cables. The connection between the HP2100S and HP2155A implements the following buses:

1. 16-bit-wide input data bus
2. 16-bit-wide output data bus
3. IO-slot address bus
4. Read and write strobes
5. Interrupt signals
6. DMA signals
7. Other status and control signals

(Interrupts can be enabled via software and are enabled on four of the IO PCAs.)

Hewlett Packard supplied several configurations of instrument IO PCAs for the HP2155A of which the AN/DPM-22 employed the following types:

1. HP12551B, relay register interface
2. HP12566B, 16-bit parallel IO interface
3. HP12604B, 32-bit input register interface
4. HP12661A, interface to HP6131 digital voltage source

Power supplies and logic circuits on PCAs do not conform to Transistor-Transistor-Logic (TTL) standards. The IO extender backplane is wire wrapped.

IO Extender Chassis Redesign

Retaining the HP2155A IO extender as the interface to the non-GPIB instrument cables was determined to be more cost-effective than creating a new instrument interface. However, the following modifications had to be made to the HP2155A.

1. Power supply. The power supply was replaced with an off-the-shelf IBM-PC tower-case-compatible power supply.

2. Backplane wiring. The wire-wrap wiring of the IO extender backplane was modified to accept the new interrupt system. This modification was easily implemented because the HP2155A backplane contains many wires that were unused in the new design. A few of these unused wires were assigned new functions.

IO Extender PCA Designs

All PCAs in the IO extender were replaced. Three of the new PCAs interface to the computer while the remainder provide the parallel digital interfaces to instruments. The new PCAs are the same size as the HP PCAs replaced. To facilitate maintenance all PCAs are double-sided; integrated circuits (ICs) are off-the-shelf TTL; and all ICs are socketed.

Computer Interface. The two PCAs and cables that connected the IO extender with the HP2100S were removed and replaced by three new PCAs and cables designed to connect with the new IOX486 PCA in the computer. The new computer interface PCAs function similarly to those in the original HP interface, except (1) DMA signals are not implemented, (2) the HP interrupt system was replaced with one compatible with the IBM-PC and 80486 microprocessor, (3) the backplane control signals were simplified for compatibility with new instrument interface PCA designs, and (4) two additional backplane address lines were defined to allow up to four IO ports on each instrument interface PCA.

Instrument Interfaces. New PCAs were designed to retain the original physical and electrical connections to the instruments. The connections to the backplane are similar to the original, except (1) the system interrupt circuitry was replaced with one compatible with the IBM-PC and 80486, and (2) the control signals for reading from and writing to the PCAs were simplified. The new 32-bit input register interface PCA was designed to be addressed as two IO ports.

SOFTWARE DESIGN

The HP Test Oriented Disk Operating System (TODS) is the operating system used with the HP2100S. Test programs were written in TODS-BASIC—an interpreted version of the BASIC language—that allowed user-developed subprograms to be linked into the language. The user-developed subprograms were written in HP2100 assembly language. Over 500 TODS-BASIC test programs and 100 HP2100 assembly language subprograms had to be analyzed and converted to an IBM-PC-compatible language.

PROGRAMMING LANGUAGE SELECTION

Microsoft Visual BASIC for DOS - Professional Version (VBDOS-PRO) was selected after evaluation of several versions of BASIC and other programming languages. The programming language selection was based on the following criteria.

1. Compatibility with HP TODS-BASIC test program syntax. VBDOS-PRO syntax is similar to that of TODS-BASIC. Maximum compatibility between the HP TODS BASIC test syntax and the new test syntax is required because nine of the 13 AN/DPM-22 test stations are not being modified. Therefore, both the old and new software configurations will be maintained. Compatibility also is required to minimize the syntax translator program that

would be used to translate HP TODS-BASIC syntax into the new programming language syntax.

2. Capability to create multimegabyte executable test program (EXE) files. When all the individual test programs and subprograms required for each test program set are linked into one MSDOS EXE file, the TPS EXE file size exceeds 1 megabyte. VBDOS-PRO supports large files with an overlay manager within the EXE file. The overlay manager moves overlay portions of the code to be executed into the lower 1 megabyte of memory and moves inactive code overlays to available memory above 1 megabyte, thereby allowing multimegabyte files to be executed.

3. Easy integration of application-specific interrupt handlers. VBDOS-PRO allows subprograms (including interrupt drivers) to be written in VBDOS-PRO or another language and linked into the test EXE file. VBDOS-PRO also allows the test program to specify which subprogram to execute when user-defined interrupt (UEVENT) occurs. Sharing data between the VBDOS and non-VBDOS software modules is easily done under VBDOS-PRO.

4. Graphical user interface is not required. Visual BASIC for Windows and other languages requiring MicroSoft Windows were considered but not selected because of the complexity of the Windows environment and the complexity of adding device drivers.

TEST PROGRAM TRANSLATION TO VBDOS-PRO

Because of the large number of existing test programs written in the TODS-BASIC programming language, automatic translation of TODS-BASIC test program source files into VBDOS-PRO syntax was essential. We created a translator program using VBDOS-PRO that identified and translated 95% of the syntax differences. Syntax differences that occurred infrequently or were difficult to detect automatically were left for manual translation. Where manual action was required, a flag was inserted in the VBDOS-PRO test program source file.

The 500-plus HP-TODS test programs were converted into 300 VBDOS-PRO source files that were compiled and linked into 10 TPS EXE files.

An eleventh EXE file, unique to the new computer, provides a menu to select tests to perform, set system configuration data, and select help displays.

ASSEMBLY LANGUAGE SUBPROGRAMS

During analysis, some of the original subprograms were determined to be unnecessary because their functions are performed by MSDOS and VBDOS-PRO. The remaining subprograms were flowcharted to understand the functions performed and the algorithms implemented. Most of those subprograms were rewritten in VBDOS-PRO, as were most new subprograms. A few subprograms were written in Microsoft MASM assembly language or Microsoft C, where VBDOS-PRO could not support a desired capability. C language

subprograms were written to control the IOX486 PCA hardware (see IO Extender Design) and respond to IOX486 interrupts.

The operator interface subprograms were completely rewritten to use the new video monitor, keyboard, and mouse. New code was written using VBDOS-PRO forms to display information to the operator.

CONFIGURATION MANAGEMENT

Software configuration during development is controlled by maintaining the master files on LAN. During the integration and test phases, the master files are downloaded into the test stations, which are all connected to the LAN. Errors identified during debugging are documented, a corrected copy of the software is tested, and the master file on the LAN is corrected by the Configuration Manager, who is the only person with Write Access to the master files.

TECHNICAL MANUALS

The cost of updating technical manuals for periodic updates traditionally has been exceedingly high and sometimes can surpass the cost of executing an Engineering Change Proposal (ECP). Updating test equipment for small changes is a common requirement, and costs over the lifetime of a test system can be extremely high when technical manuals also are updated. These costs are of common concern to most sponsors. One way to reduce the cost of documentation when updating test equipment is to have the technical manuals in magnetic media.

The engineering team was challenged to find an alternative that would achieve long-term cost savings not initially prohibitive. Various Microsoft Windows-compatible software packages were researched to determine which would be the best solution for the Navy. Some of the packages researched included Ventura Publisher, Interleaf, World View, and Interactive Authoring and Display System (IADS). The software chosen was IADS.

Most of the original text in the technical manuals was already in digital format with the drawings in paper media. The text was translated from its previous format to IADS compatibility. The text was arranged using the Definition Type Document (DTD) format developed by the Army. Drawings were scanned into CALS Group 4 format and then integrated with the text to create the new digital media technical manuals.

Technical manuals were organized according to function, broken down to subfunctions, and further broken down as required (Figure 4). Each subfunction was linked to its higher level corresponding subject using hotword technology. Drawings were linked directly to their subject. The new technical manuals were organized to be user friendly and easy to use, either on the station control computer system, or on a separate inexpensive computer and printer on a mobile cart.

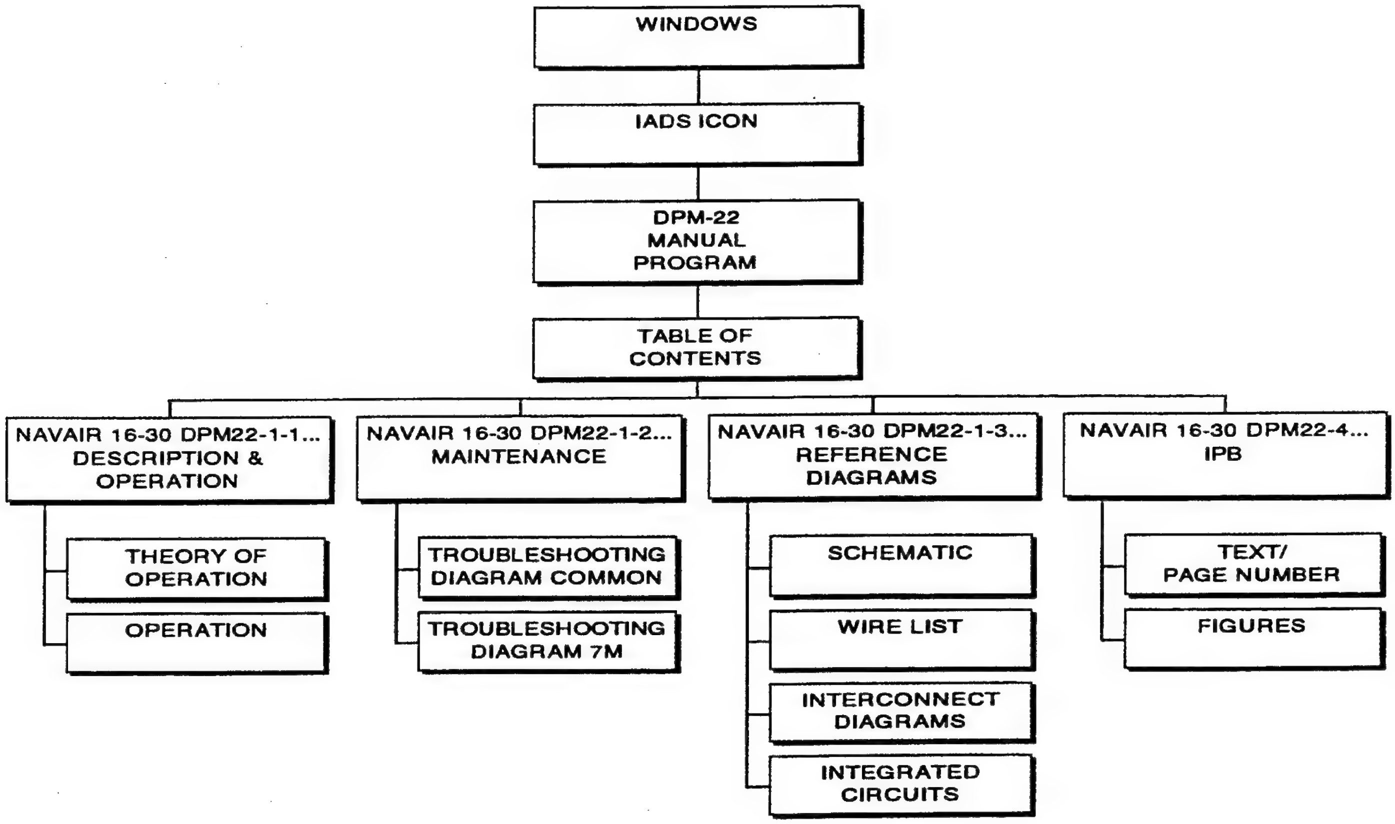

FIGURE 4. IADS Flow Diagram.

INTERACTIVE AUTHORING AND DISPLAY SYSTEM

Interactive Authoring and Display System (IADS) is a Microsoft Windows application capable of running on any computer that supports Microsoft Windows Version 3.0 or later. The software includes an authoring environment to create IADS publications and a reader-only environment for end users. IADS features hypertext and graphics links, and graphics zoom capability. An IADS-compatible document uses Standard Generalized Markup Language (SGML), ISO 8879 "tags" within text files to identify display font, color, hypertext, etc. IADS includes a control panel that allows the user to perform common operations, such as string search, page forward/reverse, and print. The control panel is accessed via the mouse. Each control panel button has a corresponding accelerator key (hotkey). Some key functions, such as the <INDEX>, <NEXT>, and <PREVIOUS> keys, are programmed by the document author. The IADS Help facility includes the hard copy manual in its entirety and is context sensitive. Thus, the help displayed is related to the current command or area. The help files provide excellent examples for an IADS author to reference when creating a document. IADS was developed by USAMICOM for the Integrated Materials Management Center (IMMC), U.S. Army, Redstone Arsenal, Ala. IADS software is available from IMMC at no

cost. The software may be used by anyone at no cost. Anybody may publish documents using IADS without any obligations to USAMICOM or anyone else.

CREATING AN IADS DOCUMENT

An IADS author creates an IADS-compatible document by adding ASCII text commands to an ASCII text version of a document. The commands (tags) determine display characteristics of each document section, such as table of contents, section headings, paragraph headings, or hotspots, to other sections or graphic images. Essentially, SGML tags are placed at the beginning and end of text areas. A document may be tagged within IADS in the "Revisable Mode" or within a users favorite editor or word processor. However, documents must be saved as ASCII text (without hidden/special characters).

Experience indicates the more expeditious way to create and modify the document is by using an editor outside of IADS. Then, load IADS, select the document, view, and repeat as necessary.

TAGGING TEXT DOCUMENTS

Tagging a document means surrounding portions of information with SGML codes. Tag codes begin with "<" and end with ">". If the tag to define the beginning of a paragraph is "<para>", then its matching end of paragraph tag would be "</para>". Essentially, the IADS author characterizes each and every portion of the document/manual by adding tags defining beginning and end of files, display frames, text font, text attributes (bold, underline, italics, color, etc.), and frame linking. There are tags to identify hypertext, and hotspot areas used to display dialog boxes, jump to other sections of a document, or display graphics images.

NOTE: Hotspot figure and table references allow the reader to quickly view each figure, zoom to a desired portion, and/or print it, then automatically return to the document.

NOTE: If tag characters "<" or ">" are inadvertently placed in some random location within the source document and not associated with actual tag names, IADS may yield erroneous messages and possibly exit back to Windows. The best way to resolve this problem is to load the suspect source file into a word processor and search for stray tags.

Sample: IADS TAGS IN A TEXT (.IDE) FILE:

```
<FILE
 TITLE="IADS Author Application - Software User's Manual"
 STYLE='iadssum.sty'
DOMAIN="author.lst">

<frame label="IADS Author Overview"
NEXT="IADS Author Overview"
PREVIOUS="IADS Author Overview"
INDEX="IADS Author TOC"
SHOWNO="NSHOWNO">
<option>
```

```
<TITLE>IADS Author Application - Overview</TITLE>

<GENDESC>Welcome to the Interactive Authoring and Display System (IADS) Author Online Help Manual. This document was developed for anyone needing assistance in utilizing any of the authoring features available in IADS. If you need assistance with the IADS Reader, the online reader documentation is also available from the Help menu.</GENDESC>

<GENDESC>The IADS authoring tools actually consist of several modules, to include IADS Author, Style Sheet, Viewimage, ZoomView, Link Verifier, and Graphics Conversion. The IADS Author application is described in this document, as well as SGML issues. The other modules are discussed in detail in separate documents.</GENDESC>

<LIST>
<HOTSPOT>Style Sheet<ACTION>
<?DISPLAY TEXT="!styles.ide"></ACTION></HOTSPOT>
<HOTSPOT>Viewimage<ACTION>
<?DISPLAY TEXT="!authview.ide"></ACTION></HOTSPOT>
<HOTSPOT>ZoomView<ACTION>
<?DISPLAY TEXT="!zoomview.ide"></ACTION></HOTSPOT>
<HOTSPOT>Link Verifier<ACTION>
<?DISPLAY TEXT="!linkverf.ide"></ACTION></HOTSPOT>
<HOTSPOT>Graphics Conversion<ACTION>
<?DISPLAY TEXT="!graphcon.ide"></ACTION></HOTSPOT>
</LIST>

<GENDESC>To find topics in the help documents use the table of contents screens and the IADS search feature. Also, look for red hotspot areas to get more information about particular topics. Use the control panel index button to return to table of contents screens, or the right mouse button to retrace your steps.
</GENDESC>
</OPTION>
</FRAME>
</FILE>
```

A screen image of tagged text is shown in Figure 5.

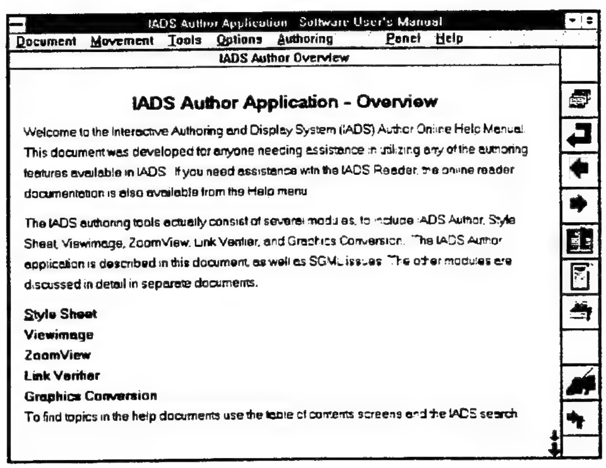

FIGURE 5. IADS Screen of Tagged Text.

EMBEDDING GRAPHICS FIGURES

Graphics files, such as figures scanned from hard copies of manuals and those readily available on file, may be incorporated into IADS documents. IADS provides two graphics display methods (display picture and zoomview) that an author must choose from at the time the document is created.

Display Picture

The display picture method displays any of four graphics file formats in their entirety. The formats are

1. Windows Device Independent Bitmap (.BMP)
2. PC Paintbrush (.PCX)
3. CCIT Group 4 (.G4, also known as .CAL)
4. Picture Definition (.PIC)

Example:

```
<?display picture="\iads\yourfig.bmp">
```

Zoomview

Zoomview is a separate utility included with IADS to allow the IADS reader to zoom to any portion of a graphics file. Zoomview is especially useful when dealing with large and/or dense graphics files, such as large assembly drawings or densely populated circuit board assemblies. File formats for Zoomview method are

1. CCIT Group 4 (.G4, also known as .CAL)
2. Picture Definition (.PIC)

Example:

```
<hotspot>
Fig 1 Sh 1-2.<action><?execute program=
"c:/iads/programs/zoomview
/iads/dpm22/dpm224/gfx/01-1.pic"></action>
</hotspot>
```

The actual graphics image file format must be .G4. The .PIC file is an optional information file used to overlay a figure title on the IADS display or identify hotspots to other figures, messages, or text sections. The .PIC file is created by the author within the Viewimage Author application.

NOTE: We used Hi-Jaak by Inset software to crop and/or convert from various graphics file formats to .CAL format. Then, we renamed the .CAL file to .G4, as required by IADS.

IADS OPERATING ENVIRONMENT

IADS provides a private user name/password feature if the document requires restricted access. Otherwise, a user would login to IADS as "guest" and leave the password blank. The screen is comprised of the following parts:

1. Main Window Bar. Displays the title of the current document at the top of IADS application window.

2. Menu Bar. Assortment of pull-down menus to access functions. The IADS author Menu Bar includes editing features unavailable to an IADS reader.

3. Frame Title Bar. Displays the title of the current frame.

4. Control Panel. Contains buttons that can be accessed by the user. The author can change any of the 11 default buttons, function, location, size, or visibility.

Control Panel Button Descriptions:

HELP
Access any help screen that the author has attached to the current frame.

INDEX
Jump to a frame that the author assigned as a parent to the current frame. Example: A frame showing a parts list index may be chosen as the frame to jump to if the <INDEX> button is pressed from within a detailed part description.

BACK
A history list of frames viewed previously in the current IADS session. To jump to a frame, simply highlight it and press the <ENTER> key.

PREV
To jump to the previous frame or link to a frame in another location of the document (programmed by author).

NEXT
To jump to the following frame or link to a frame in another location of the document (programmed by author).

BOOKMARK
Mark a spot for easy return.

NOTE
To place private or public notes in a frame, such as warnings or hints related to frame.

PRINT
Print the current frame.

REPORT
A problem report form to identify problems with IADS to be forwarded to IMMC for review.

SEARCH
Search for text string in area defined by user.

EXIT
Exit IADS.

5. Display Area. Contains the contents of the current frame, both text and graphics. The mouse cursor usually appears as an arrow, except when it is over a hotspot where it appears as a hand with a pointing finger.

SYSTEM REQUIREMENTS

Minimum requirements include the following:

PC using an 80386 microprocessor
Four (4) megabytes of RAM
Microsoft Windows version 3.0
One 3.5-inch (1.44-megabyte) disk drive
Ten (10) megabytes of available hard disk space
Microsoft mouse or compatible pointer device
Microsoft MS-DOS 5.0

Recommended requirements include the following:

PC using an 80486 microprocessor
Eight (8) megabytes of RAM
Microsoft Windows version 3.1

SYSTEM LIMITATIONS

Frames per file		32,000
Tag name length		32 characters
Attribute name length		8 characters
Attribute value length		240 characters
Exceptions:	Frame labels	64 characters
	Frame titles	128 characters
Attributes per tag		16
Text hotspots per frame		95

Hotspots per graphic	104
File specification length	68 characters
Entries per stylesheet file	256 tag records

IADS UTILITIES

IADS includes the following utilities:

1. IADS Author. Application to create IADS documents providing text editing, hypertext linking, and graphics hotspotting capabilities.

2. IADS Reader. (The primary user application). A read-only environment for an engineer or technician in the field using the IADS electronic publication to assist in the resolution of a problem.

3. Viewimage Author. To associate a picture information file (.PIC) with a CALS (.G4) graphics file. The author may assign a (frame) title to be displayed when image is viewed in addition to creating hotspots at any place on the graphics image.

NOTE: NAWCWPNS has placed hotspots on multisheet (graphics) figures to quickly jump forward or back in a series.

4. Viewimage Reader. A view-only environment for an end user to view an image directly without accessing the publication and frame where it is hotspotted from.

5. Stylesheet. Application to control the appearance or formatting of an IADS document. The author can associate formatting properties with the SGML tags in his document without actually changing the document tags or content.

NOTE: The Stylesheet application works well to quickly view the effect of different tag properties, much like a "what-if" environment. When the desired tag properties are determined, the change may be saved.

6. Zoomview. This application is a raster graphics display tool that accepts CALS Group 4 (.G4) files. Each image can be loaded so that all or part is visible. Once loaded, Zoomview provides image functions, such as zooming, scrolling, panning, printing, and marking. This utility is generally accessed automatically via a hotspot that an author would program into the publication.

7. Link Verifier. A tool that checks to see if all the document hyperlinks, including ones inside PIC files, can be found on the current document.

8. Graphics Conversion. A tool to convert graphics files from file types .G4, .MAC, .PCX, or .XRX to any of file types .BMP, .G4, .IMG, .PCX.

IADS DPM22 EXAMPLES

The DPM22 automatic test station technical manuals were converted from hard paper copies to a computer display copy using IADS. This computer display system can be used by the DPM22 station repair technicians to repair the station without the need for paper schematics. Examples of the computer display system are shown in Figures 6 through 10.

The DPM22 software manual created on IADS is based partially on descriptive information imbedded in each component source file header block. With 1000-plus files and the likelihood of software updates, we needed an automated method to extract, sort, and insert IADS tags in source file header text for immediate placement into the paperless manual. A software program was developed to perform these tasks. Figure 10 is an example of the output from the generated files.

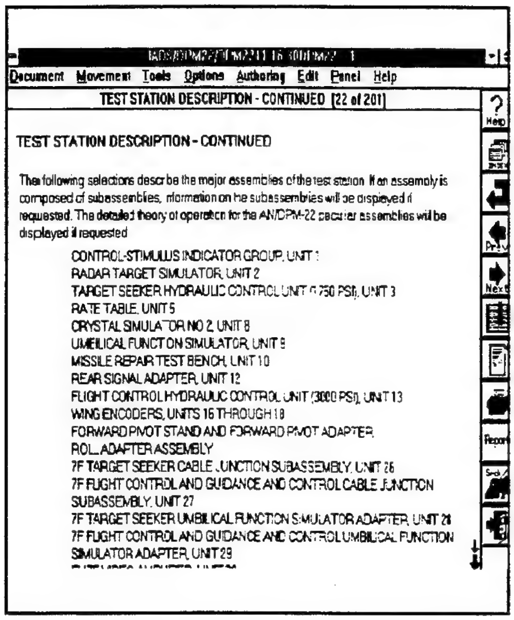

FIGURE 6. DPM22 Text.

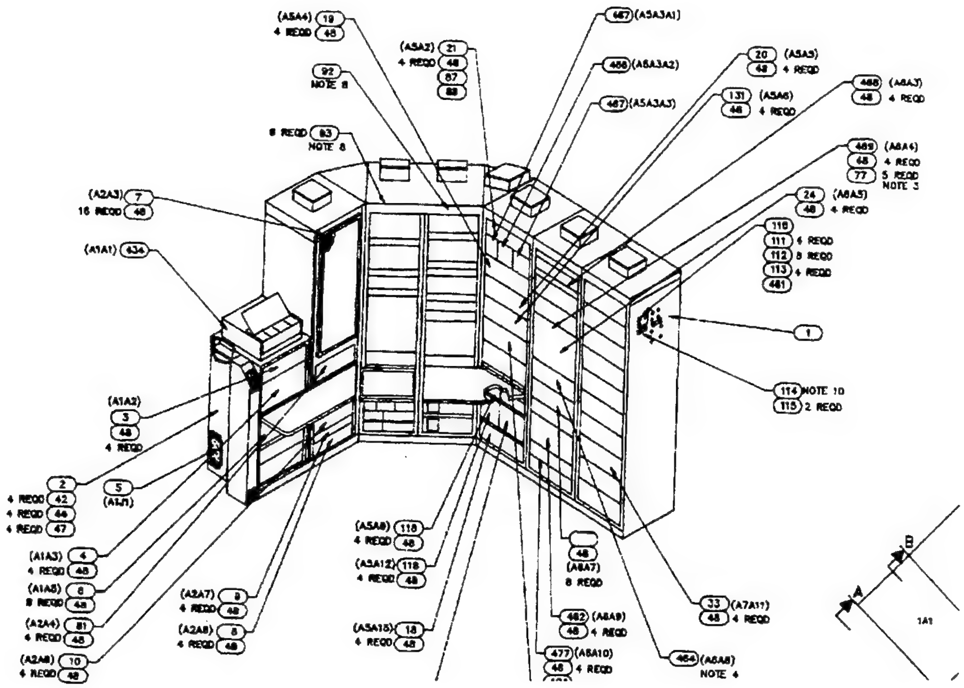

FIGURE 7. Part Breakdown Diagram.

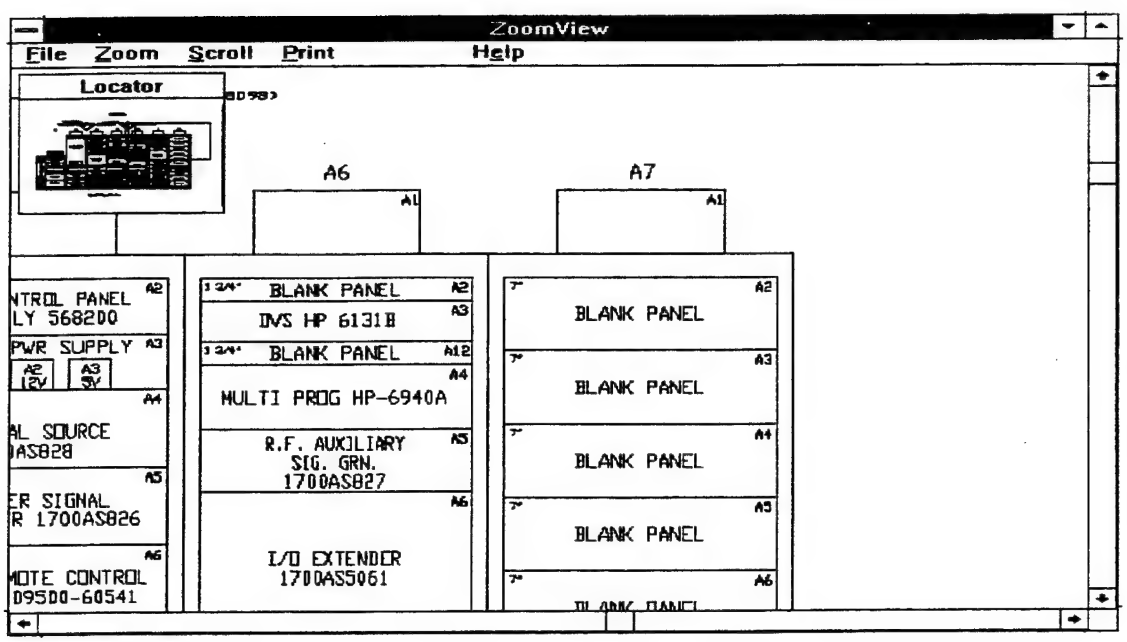

FIGURE 8. Zoomview of Component Location Diagram.

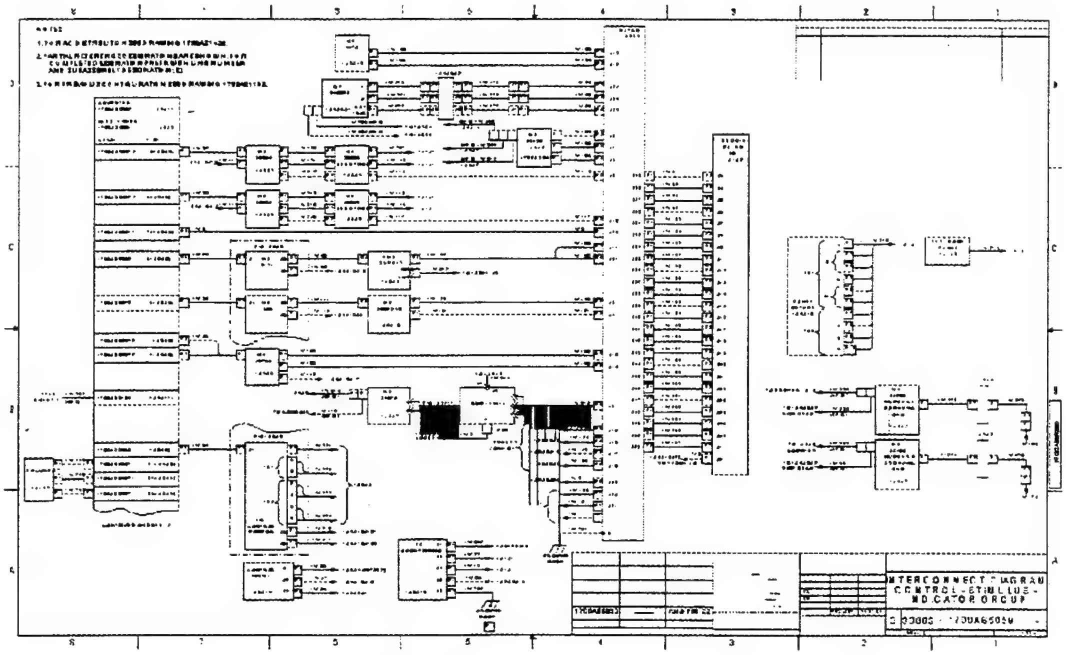

FIGURE 9. Electrical Schematic.

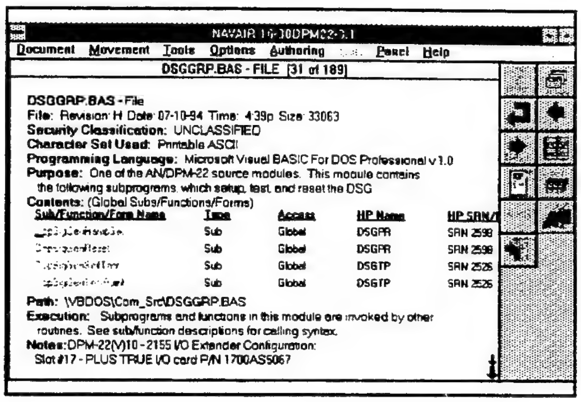

FIGURE 10. Header Text.

PROJECT MANAGEMENT

The Navy took an innovative approach to decreasing life-cycle costs while addressing the testability requirement. By using cost control and performance measurement techniques practiced in the private sector, the difficulty of measuring program status can be monitored effectively. The Navy developed cost-performing tools to streamline technical program management and focus resources on the engineering tasks.

MANAGEMENT TOOLS

The project requirements were assessed by the engineering team at Point Mugu and a Work Breakdown Structure (WBS) was developed along with a WBS dictionary. The assessment included an estimate of each lower level task with a schedule of completion for each task. A detailed manloading chart by month per task was used as a guide to control costs that would meet schedule requirements. Where appropriate, the WBS was broken down to as much as six levels. Level 0 describes the overall program requirements. Level 1 was broken into the nonrecurring and recurring parts of the program. The Level 2 nonrecurring part was divided into four sections: (1) management and analysis, (2) test station upgrade, (3) TPS development, and (4) integrated logistics support. The recurring part at Level 2 was divided into two categories of management and production for the retrofitting of additional test sets. The four sections from Level 2 were further broken down into lower levels to fully describe the various tasks associated with each level and to closely monitor progress. Figure 11 illustrates the WBS to six levels.

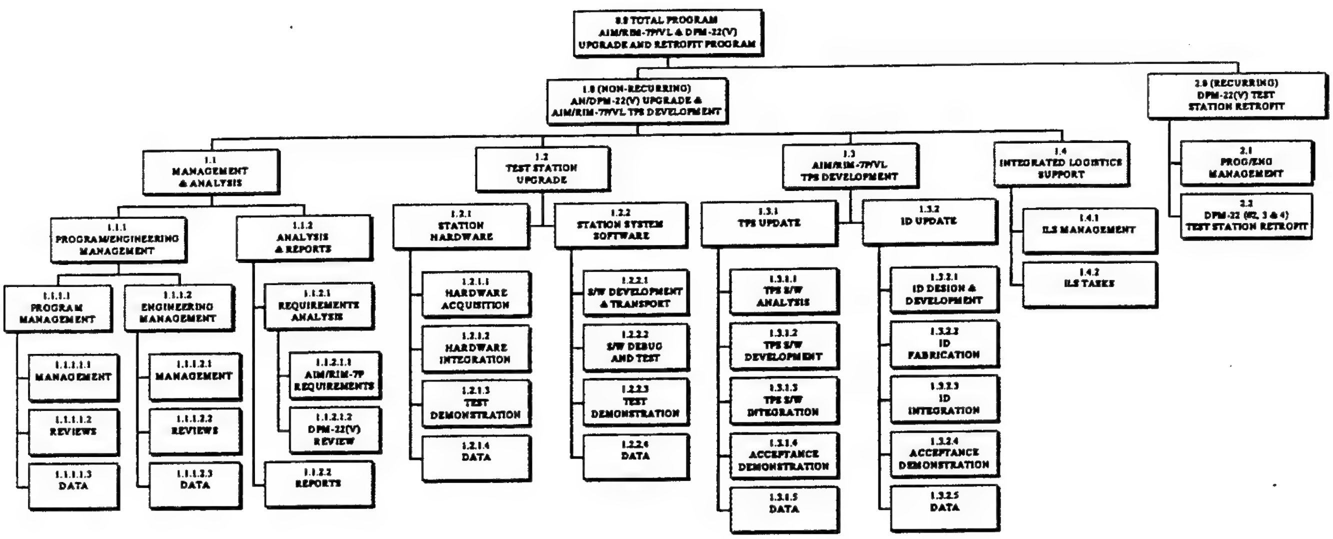

FIGURE 11. Work Breakdown Structure.

Using the WBS as a guideline, a personnel organizational structure was created. This structure served two purposes: (1) identified individual responsibility, and (2) linked the WBS task elements to personnel according to their capability. Work orders (WOs) were developed to assign individual responsibility. The WOs tied the WBS structure to the organizational chart and completed the loop. The WO described the work required in accordance with the WBS dictionary, the period of performance according to schedule requirements, the estimated man-hours, and the deliverables required on both a periodic basis and at completion. Figure 12 shows a typical WO.

WORK ORDER

No. 1311

Timecard Charge No. 325WSPB6 Date: 8/1/93 Rev: ORIG

Program: SPARROW/DPM-22/AIM/RIM UPGRADE WBS# 1.3.1.1

Package Title: AIM/RIM-7P TPS S/W ANALYSIS

Assigned To: Paul McMartin Phone: (805) 989-8736

Schedule Start: 8/1/93 Finish: 1/31/94

Budget Labor Hours: 1,300 ODC $0.00

Work Package Description:

1. Review Systems Engineering/Analysis Report on AIM/RIM-7P Vertical Launch (VL).
2. Support TPS SDR and PDR Meetings.
3. Review NWC-2410 Specification and analyze S/W implementation approach to test AIM/RIM-7P VL TPS on upgraded DPM-22(V) Test Station.
4. Coordinate approach with WBS 1.3.2.1 TPS ID H/W Design & Development task.
5. Report personnel hours, by named persons, and percent complete at WBS level 1.3.1.1 to the Engineering Manager, Elden Sandy, on a weekly basis. Highlight any problems that could have a cost or schedule impact on your tasks.

Deliverables:

1. Weekly Reports to Engineering Management.*
2. SDR and PDR Engineering Presentation Material.**
3. Definition of optimum AIM/RIM-7P VL TPS S/W approach and implementation methods that meet NWC-2410 requirements.

*Data Preparation Task 1.3.1.5 provides clerical support.

** 1. Review Task 1.1.2.2 covers SDR and PDR attendance.
2. Coordinate SDR and PDR preparation with Engineering Management (WBS 1.1.1.2.2).

Program Mgr: Jesse Zapata W.O. Mgr: Paul McMartin

FIGURE 12. Work Order.

Each WO was discussed with responsible personnel for complete understanding of the task requirements before signing the WO to acknowledge the commitment. Monthly status reports of each WBS element from the engineering team were required. The data in these reports were used as inputs into a program management software package, Performance Assessment Routine (PAR), to help determine the status of the program. To monitor progress, a budget was developed using the manloading chart (Figure 13) as the basis of cost and performance expectation.

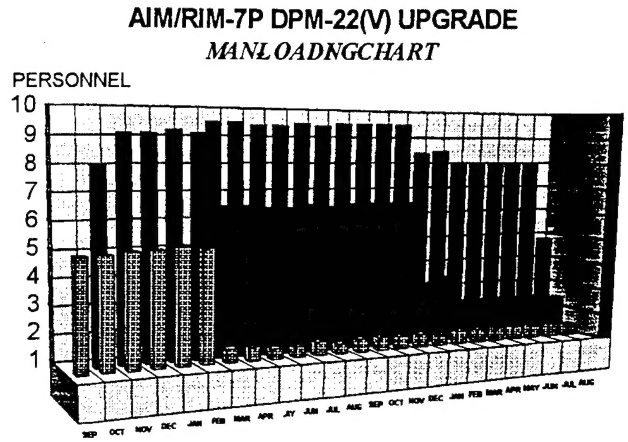

FIGURE 13. Man-Loading Chart.

The monthly engineering team status report was used by program management in conjunction with PAR to generate data used to evaluate the status of the project. The PAR program takes lower level data inputs (actual expenditure and percent complete) and compares the actual cost to a previously budgeted estimate. PAR generates cost and schedule variance and performance index data. This process is continued until all results are known at the lowest level. The data generated at the lowest level are then input at the next higher level and PAR generates cost and schedule variance at this level. The process is continued at each higher level until a top level cost and schedule variance and performance data are derived. Figure 14 shows a typical PAR input data form. The task element estimates of manloading data (in dollar values), generated as a result of the WBS, are pre-requisite inputs to PAR at all WBS levels for each reporting period.

Cost Variables	Month								
	1	2	3	4	5	6	7	8	N
$ BUDGET									
$ EXPEND									
% COMPL									
$ BCWS									
$ ACWP									
$ BCWP									
CSTVAR									
SCHVAR									

FIGURE 14. Performance Assessment Routine.

A monthly report is prepared for the NAVAIR sponsor using the data generated by PAR. The report is a five-part report and includes executive summary, cost reports down to WBS Level 2, top-level expenditure profile versus current funding plot, cost and schedule variance plot, schedule status, and WBS format reports. These monthly reports have provided an in-depth project cost/performance visibility to the sponsors that have rarely been seen.

PROJECT STATUS

All hardware has been designed, procured, and assembled. All software coding has been completed. Hardware and software have been integrated. Subprogram software has been debugged. Test program software debugging is nearing completion. IADS technical manuals have been developed.

CONCLUSION

The use of management tools and processes is an essential element to controlling costs and providing feedback to the sponsor on the program status. The use of COTS technology to enhance the performance of aging equipment and provide for future program requirements is critical to sponsors as DOD budgets decrease. Test station life-cycle costs can be reduced if alternative methods to updating the documentation are viewed from a total programmatic point of view. The use of IADS to digitally represent the technical manuals for the multipurpose AN/DPM-22 test station has reduced future update costs. The tools and methods used in this program provide the engineering team and program management with constructive feedback on the status of the program.

RECOMMENDATIONS

This modification process can be adapted to upgrade the computers of other HP9500 automated test systems. The following processes are recommended:

1. Replace/modify computer hardware as described. Hardware is either COTS or assembled from complete design documentation.

2. Translate test software from the TODS-BASIC syntax to VBDOS-PRO syntax using a variation of the translator program developed for the AN/DPM-22.

3. Recode test-set-unique assembly language subprograms in VBDOS-PRO or other MSDOS compatible language. However, some generic HP9500 subprograms will not have to be recoded because they are unnecessary or replacements have already been developed.

4. Minimize management costs by developing management tools with commercially available software.

REFERENCE

1. U.S. Army Missile Command. Software Users Manual, Interactive Authoring and Display System. Redstone Arsenal, Ala. (AMSMI-MMC-BM-A, Document UNCLASSIFIED.)